UCRL- 93503 Revision 1
PREPRINT

ODRIC, a One-Dimensional Linear Resistive MHD Code in Cylindrical Geometry

A.A. Mirin
R.J. Bonugli
N.J. O'Neill
J. Killeen

This paper was prepared for submittal to
Computer Physics Communications

February 1986

Lawrence
Livermore
National
Laboratory

ODRIC, a One-Dimensional Linear Resistive MHD Code in Cylindrical Geometry*

A.A. Mirin, R.J. Bonugli, N.J. O'Neill and J. Killeen
National Magnetic Fusion Energy Computer Center
Lawrence Livermore National Laboratory
Livermore, CA 94550

PROGRAM SUMMARY

Title of Program: ODRIC

Catalogue number:

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland

Computer for which the program is designed and others on which it is operable:

Computer: CRAY-1,X-MP Installation: National Magnetic Fusion Energy Computer Center, Lawrence Livermore National Laboratory

Operating system or monitor under which the program is executed: CTSS

Programming language used: FORTRAN

---

*Work performed under the auspices of the U.S.D.O.E. by Lawrence Livermore National Laboratory under contract W-7405-ENG-48.

High speed storage required: 97,000 + 335 * J (real) words,
140,000 + 632 * J (complex) words,
where J is the number of meshwidths.

No. of bits in a word: 64

No. of lines in combined program and test deck: 9800

Keywords: One-Dimensional, Resistive, Implicit, Linear, MHD

Nature of Physical Problem

The primitive linearized resistive MHD equations in cylindrical geometry are advanced in time. Separate equations for the electron and ion temperature perturbations are solved. Hall terms and the thermal force vector are included in Ohm's law. Anisotropic thermal conductivity and viscosity are included in the model. The plasma is assumed to be quasineutral. Fourier analysis in the poloidal and toroidal directions is performed, resulting in a 1-D (radial) system for the perturbed quantities.

Method of Solution

A finite difference method is used. Spatial derivatives are approximated by central differences. The mesh need not be uniform. Temporal differences are fully implicit.

Restrictions on the Complexity of the Problem

The user must adjust the size and type of radial mesh, in particular in the singular layer, in order to model the physics accurately. For nonaxisymmetric perturbations ($m \neq 0$; see Eq. (21)) a small amount of viscosity is necessary to prevent numerical instability. This is discussed further.

Typical Running Time

Two versions of the code are provided (see Sect. 1 of long write-up). The real version takes 40 μs per meshpoint per timestep on the Cray-1. The complex version takes 113 μs per meshpoint per timestep.

References

[1]. A.A. Mirin, N.J. O'Neill, J. Killeen, R.J. Bonugli and M.J. Ellis, "Linear Studies of Resistive Interchange Modes in a Cylindrical Reversed Field Pinch," to appear in Phys. Fluids.

Note: Separate source codes for the real and complex versions are included in the test deck. Test problem 1, which uses the real version, is followed by test problem 2, which uses the complex version. To run the compile-load-go deck at NMFECC, the sequence numbers in columns 77-80 must be removed from the job control statements. These commands, which have an asterisk in column 1, occur on lines 1, 2, 4666-4668, 4742-4746, 9641-9643, 9717-9719.

LONG WRITE-UP

1. Introduction

Determination of the MHD stability of a particular equilibrium generally requires a numerical approach. As a first step it is often beneficial and therefore quite common to perform a linear normal mode analysis. One way of doing this is to convert the time-dependent system of equations into an eigenvalue problem, to discretize that set of equations to form a matrix eigenvalue problem, and then to solve that resulting system for the growth rates and mode structures. This technique has been used in the well-known codes PEST[1] and ERATO[2]. Another approach is to integrate directly the time-dependent equations to obtain the fastest growing mode. Such a procedure has been carried out by Dibiase and Killeen[3] and by Shestakov, et al.[4]. This latter method is often more tractible for complicated systems such as those in which transport effects, e.g. resistivity and thermal conductivity, are included.

The model presented here is a one-dimensional (radial), linear initial value code in cylindrical geometry. The compressible, resistive MHD equations are integrated in time until the fastest growing mode emerges. Separate equations for the electron and ion temperature perturbations are solved. Hall terms and the thermal force vector are included in Ohm's law. Anisotropic thermal conductivity and viscosity are included in the model. Quasineutrality is assumed.

The code uses complex arithmetic. In the absence of certain physical effects (e.g. Hall and thermal force terms, transverse thermal conductivity, FLR terms), the eight complex equations decouple into two identical real

systems of eight equations each. Since in this case one need solve only a single real system, two versions of the code are provided - a complex version and a real version. The demonstration test problems, which solve for resistive interchange modes in a reversed field pinch, use both the real and complex versions of the code.

2. Basic Equations

In nondimensional form (the normalization of variables is summarized in Table 1), the equations of interest are as follows:

$$\frac{\partial \rho}{\partial t} + \nabla\cdot(\rho \underline{v}) = 0 \tag{1}$$

$$\frac{\partial T_e}{\partial t} + \underline{v}\cdot\nabla T_e = -(\gamma-1)\, T_e\, \nabla\cdot\underline{v} + \frac{2(\gamma-1)}{\rho}\, \underline{J}\cdot(\underline{E} + \underline{v}\times\underline{B})$$

$$+ \frac{(\gamma-1)}{\rho}\nabla\cdot(\underline{\kappa}_e\, \nabla T_e) + \nu_{ei}\,(T_i - T_e) \tag{2}$$

$$\frac{\partial T_i}{\partial t} + \underline{v}\cdot\nabla T_i = -(\gamma-1)\, T_i\, \nabla\cdot\underline{v}$$

$$+ \frac{(\gamma-1)}{\rho}\,\nabla\cdot(\underline{\kappa}_i\, \nabla T_i) + \nu_{ei}\,(T_e - T_i) \tag{3}$$

$$\frac{\partial \underline{v}}{\partial t} + \underline{v}\cdot\nabla\underline{v} = -\frac{1}{2}\nabla T - \frac{T}{2\rho}\nabla\rho + \frac{1}{\rho}\,\underline{J}\times\underline{B} + \frac{1}{2\rho}\,\nabla\cdot\underline{\underline{\pi}} \tag{4}$$

$$\frac{\partial \underline{B}}{\partial t} = - \nabla \times \underline{E} \tag{5}$$

with the auxiliary conditions

$$\underline{J} = \nabla \times \underline{B} \tag{6}$$

$$\underline{E} = - \underline{v} \times \underline{B} + \eta \underline{J} + \underline{E}_{Hall} + \frac{1}{\rho} \underline{R}_T \tag{7}$$

$$\underline{E}_{Hall} = \frac{\nu}{\rho} \left[\underline{J} \times \underline{B} - \frac{1}{2} \nabla(\rho T_e) \right] \tag{8}$$

The normalized resistivity η (which is taken to be constant) is the inverse Lundquist number $S = \tau_R/\tau_A$, the ratio of the resistive diffusion time to the Alfven transit time. The plasma is assumed to be hydrogenic and quasineutral ($n = n_e = n_i$, hence $\rho = m_i n$), and the quantities $T_e/(\gamma - 1)$ and $T_i/(\gamma - 1)$ are actually the electron and ion energies per ion mass, so that the total energy-density is $m_i nT/(\gamma - 1)$, where $T = T_e + T_i$. The thermal conductivity tensors (defined as m_i times the usual definition [5]) $\underline{\underline{\kappa}}_e$ and $\underline{\underline{\kappa}}_i$ are given according to

$$\underline{\underline{\kappa}}_e \cdot \nabla T_e = \kappa_{e\parallel} \nabla_\parallel T_e + \kappa_{e\perp} \nabla_\perp T_e + \kappa_{e\times} \underline{b} \times \nabla T_e \tag{9}$$

$$\underline{\underline{\kappa}}_i \cdot \nabla T_i = \kappa_{i\parallel} \nabla_\parallel T_i + \kappa_{i\perp} \nabla_\perp T_i - \kappa_{i\times} \underline{b} \times \nabla T_i \tag{10}$$

where $\underline{b}$ is a unit vector in the direction of the magnetic field, $\nabla_\parallel f = (\nabla f \cdot \underline{b})\underline{b}$ and $\nabla_\perp f = \nabla f - \nabla_\parallel f$. The quantity γ is the ratio of specific heats. The viscous stress tensor $\underline{\underline{\pi}}$ is expressed in terms of the Navier-Stokes tensor W by means of the five viscosity coefficients:

$$\pi = \mu_o W_o + \mu_1 W_1 + \mu_2 W_2 - \mu_3 W_3 - \mu_4 W_4 \tag{11}$$

where

$$\begin{aligned}
W_{oij} &= \frac{3}{2}\left(b_i b_j - \frac{1}{3}\delta_{ij}\right)\left(b_k b_\ell - \frac{1}{3}\delta_{k\ell}\right) W_{k\ell} \\
W_{1ij} &= \left(\delta^{\perp}_{ik}\delta^{\perp}_{j\ell} + \frac{1}{2}\delta^{\perp}_{ij}\, b_k b_\ell\right) W_{k\ell} \\
W_{2ij} &= \left(\delta^{\perp}_{ik} b_j b_\ell + \delta^{\perp}_{j\ell} b_i b_k\right) W_{k\ell} \\
W_{3ij} &= \frac{1}{2}\left(\delta^{\perp}_{ik}\,\epsilon_{jm\ell} + \delta^{\perp}_{j\ell}\,\epsilon_{imk}\right) b_m\, W_{k\ell} \\
W_{4ij} &= \left(b_i b_k\,\epsilon_{jm\ell} + b_j b_\ell\,\epsilon_{imk}\right) b_m W_{k\ell}
\end{aligned} \tag{12}$$

with

$$W = \nabla \underline{v} + (\nabla \underline{v})^t - \frac{2}{3}\delta_{ij}\,\nabla\cdot\underline{v} \tag{13}$$

Here, $\delta^{\perp}_{ij} = \delta_{ij} - b_i b_j$ and ϵ_{ijk} is an antisymmetric unit tensor. The thermal force term $\underline{R}_T$ is given as

$$\underline{R}_T = -\gamma_T\, \rho\, \nabla_{\parallel} T_e, \tag{14}$$

where terms proportional to $\nabla_{\perp} T_e$ and $\underline{b} \times \nabla T_e$ have been ignored. The Hall parameter ν is simply $(\omega_{ci}\tau_A)^{-1}$, where ω_{ci} is the ion cyclotron frequency and τ_A is the Alfven time.

An attempt has been made to include most of the relevant physics contained in Braginskii's[5] theory. Tensor representations of the viscosity and thermal conductivities are utilized due to the vast disparity of values in the perpendicular and parallel directions. The resistivity is treated as a scalar because of the mere factor of two difference between perpendicular and parallel values. Ohm's law neglects electron inertia and viscosity but contains both Hall terms and the thermal force vector $\underline{R}_T$; the $\nabla_\perp T_e$ component of $\underline{R}_T$ has been ignored since it is of order $(\omega_{ce}\tau_e)$ smaller than the $\underline{b} \times \nabla T_e$ term (of $\underline{R}_T$), and that term is dropped since it is of order $(\omega_{ce}\tau_e)$ smaller than the ∇p_e term in $\underline{E}_{Hall}$; here ω_{ce} is the electron cyclotron frequency and τ_e is the electron collision time. Although they have a small effect, similar higher order terms are included in the viscous and thermal conduction terms, more as a matter of history than design. The component of the heat flux due to friction, q_u, estimated to be a few orders of magnitude smaller than the thermal heat flux for contemporary magnetic fusion devices, has been ignored. Viscous heating, which goes as velocity squared, is not included since in the absence of equilibrium flow it is a nonlinear effect.

It is acknowledged that even after all of these considerations, the code model most likely does not truly represent the physics. This is because the physics is most likely not classical due, for instance, to the long mean free path parallel to the magnetic field. Provision is made, however, to multiply the classical values by enhancement factors if desired. Further details may be found in Mirin, et al.[6].

3. Linearization

Cylindrical coordinates (r,ϕ,z) are used. Each dependent variable is written as a sum of a zeroth order and first order component, the former depending only on r. Eqs. (1-5) are then linearized about the zeroth order state. (A further assumption of the linearization is that the transport coefficients depend only on zeroth order variables.) Setting $\underline{v}_o = 0$ results in the following equations:

$$\rho = \rho_o \qquad \text{(independent of r)}$$

$$\frac{\partial T_{eo}}{\partial t} = \frac{2(\gamma-1)}{\rho_o}\frac{\eta}{S}\underline{J}_o\cdot\underline{J}_o + \frac{(\gamma-1)}{\rho_o}\nabla\cdot(\underline{\kappa}_e\nabla T_{eo}) + \nu_{ei}(T_{io} - T_{eo}) \tag{15}$$

$$\frac{\partial T_{io}}{\partial t} = \frac{(\gamma-1)}{\rho_o}\nabla\cdot(\underline{\kappa}_i\nabla T_{io}) + \nu_{ei}(T_{eo} - T_{io}) \tag{16}$$

$$\rho_o\nabla T_o = 2(\nabla\times\underline{B}_o)\times\underline{B}_o \tag{17}$$

$$\frac{\partial \underline{B}_o}{\partial t} = -\nabla\times(\eta\,\nabla\times\underline{B}_o) \tag{18}$$

with $\underline{B}_o = \hat{\phi}\,B_{\phi o}(r) + \hat{z}\,B_{zo}(r)$ and $T_o = T_{eo} + T_{io}$; note that ρ_o has been taken to be independent of r. The factor of two in Eq. (17) is due to the normalization. The quantities T_{eo}, T_{io} and $\underline{B}_o$ will change very slowly since S is large and $\nabla_\parallel = 0$. Thus, the radial component of Eq. (17) together with non-changing T_{eo}, T_{io} and B_o defines an approximate equilibrium which may be thought of as constant in time when analyzing sufficiently fast-growing perturbations. It is such an equilibrium which is used.

Temporal equations for the perturbed variables are obtained by ignoring nonlinear combinations of first order quantities. For example, setting

$\rho = \rho_o + \rho_1$ and $\underline{v} = \underline{v}_o + \underline{v}_1$ in Eq. (1) and noting that ρ_o is independent of r and t and $\underline{v}_o = 0$ yields the equation:

$$\frac{\partial \rho_1}{\partial t} = - \nabla \cdot (\rho_o \underline{v}_1) \tag{19}$$

The equation for the perturbed electron temperature provides a more complicated example. For ρ_o independent of r, and T_o constant in time, Eqs. (2, 6-8, 11-14) yield the following:

$$\begin{aligned}\frac{\partial T_{e1}}{\partial t} = & -\underline{v}_1 \cdot \nabla T_{eo} - (\gamma-1) T_{eo} \nabla \cdot \underline{v}_1 + \frac{(\gamma-1)}{\rho_o} \Bigg\{ 2(\nabla \times \underline{B}_o) \cdot \Bigg[\eta \nabla \times \underline{B}_1 \\ & + \frac{\nu}{\rho_o} \left[(\nabla \times \underline{B}_o) \times \underline{B}_1 + (\nabla \times \underline{B}_1) \times \underline{B}_o - \frac{1}{2} \nabla (\rho_o T_{e1} + \rho_1 T_o) \right] \\ & - \gamma_T \left[\nabla T_{e1} \cdot \underline{b}_o + \nabla T_{eo} \cdot \underline{b}_1 \right] \cdot \underline{b}_o \Bigg] + 2(\nabla \times \underline{B}_1) \cdot \left[\eta \nabla \times \underline{B}_o + \frac{\nu}{2} \nabla T_{io} \right] \\ & \nabla \cdot (\underline{\kappa}_e \nabla T_{e1}) \Bigg\} + \nu_{ei} (T_{i1} - T_{ei})\end{aligned} \tag{20}$$

Equations for the other perturbed variables are obtained in an analogous manner.

The perturbed quantities are represented as Fourier modes

$$U_1(r,\phi,z,t) = \hat{U}(r,t) \exp[i(m\phi+nkz)], \tag{21}$$

where k is the inverse aspect ratio of the corresponding torus. A system of eight complex-valued second order partial differential equations, written in

the form $\frac{\partial \hat{U}}{\partial t} = L(\hat{U})$, is solved numerically for the quantities $\hat{\rho}$, $\hat{T}_e$, $\hat{T}_i$, $\hat{V}_r$, $\hat{V}_\phi$, $\hat{V}_z$, $\hat{B}_r$ and $\hat{B}_\phi$; $\hat{B}_z$ is obtained from $\nabla \cdot \underline{B} = 0$ (hence $n \neq 0$).

4. Boundary Conditions

At $r = 0$ regularity conditions identical to those of Dibiase and Killeen[3] are used. All perturbed variables are zero except for $\hat{\rho}$, $\hat{T}_e$, $\hat{T}_i$, $\hat{V}_z$, if $m = 0$ and $\hat{V}_r$, $\hat{V}_\phi$, $\hat{B}_r$, $\hat{B}_\phi$ if $m = 1$, which instead have zero first derivative (in r). (Note that in order to guarantee unique Cartesian components of the $m = 1$ variables, it is necessary to have $\hat{V}_r + i\hat{V}_\phi = 0$ and $\hat{B}_r + i\hat{B}_\phi = 0$; this condition (which over-specifies the discretized problem) is not invoked.) At $r = r_{max} = r_{wall}$, free-flow and conducting wall boundary conditions combined with pressure balance are implemented. (Since magnetic fusion experiments often have an outer conducting wall, those conditions are often appropriate when modeling resistive instabilites.) The quantities V_r and B_r are zero, and the quantities ρ, T_e, T_i, rV_ϕ, V_z and rB_ϕ have zero first derivative.

5. Numerics

The "hatted" variables defined in Eq. (21) are represented on a radial mesh which is allowed to be nonuniform to allow for their possibly localized behavior. Centered radial differencing is used, and a fully implicit difference scheme is implemented. This results in the block tridiagonal system

$$-A_j \hat{U}_{j+1}^{n+1} + B_j \hat{U}_j^{n+1} - C_j \hat{U}_{j-1}^{n+1} = D_j \quad . \tag{22}$$

Eq. (22) is solved according to the method of Richtmyer and Morton[7]:

$$E_j = (B_j - C_j E_{j-1})^{-1} A_j, \quad j \geq 1 \tag{23}$$

$$G_j = (B_j - C_j E_{j-1})^{-1} (D_j + C_j G_{j-1}), \quad j \geq 1 \tag{24}$$

followed by

$$\hat{U}_j^{n+1} = E_j \hat{U}_{j+1}^{n+1} + G_j, \quad j < J \tag{25}$$

where the mesh index runs from 0 to J. The equations are time-integrated until the fastest growing mode emerges. An algorithm of Buneman[8] is used to analyze the complex growth rate $\gamma + i\omega$ (growth being defined according to $\exp[\gamma + i\omega)t]$.

For nonaxisymmetric modes ($m \neq 0$) a small amount of viscosity is necessary to prevent numerical instability. This instability has to do with the fact that in the absence of viscosity, velocity values at neighboring meshpoints are coupled only through the other variables. Because the force terms are largest in the radial direction, it is the $\hat{V}_r$ equation which is most susceptible to this instability. The amount of viscosity required is an approximately linearly increasing function of S and a quadratically decreasing function of meshwidth, and is problem dependent. For a recent study of resistive interchange modes in a reversed field pinch[6], numerical stability was attained with values of viscosity equal to 1×10^{-4} times classical for cases having 1500 meshpoints, m equal to 1 and S equal to 1000.

For situations in which it is desired to reduce the higher wave numbers in the solution, there is an option to artificially smooth each perturbed variable through addition of an undivided second difference. This option usually does not have to be invoked, since typically the only "troublesome" variable is $\hat{V}_r$, whose smoothing may be handled through physical viscosity. Whether using artificial smoothing or the actual viscosity term to smooth calculations, the use of too large a parameter will result in a nonphysical reduction in the growth rate.

Convergence checks have been carried out in various regions of parameter space, in which the timestep Δt and the number of meshpoints J have been varied. The criterion of less than a percent change in the growth rate when doubling J and halving Δt has been adhered to. Most of the cases have been carried out with J = 1000, although some of the higher S cases have been checked with J as high as 3300. The results have generally been insensitive to Δt provided $\gamma \Delta t < 0.05$.

The code contains built-in checks to make sure that the implicit discretization of each term is being performed correctly. That is, the right-hand side of each equation may be differenced <u>explicitly</u> at the end of the timestep and compared with the <u>backward</u> temporal difference of the left-hand side, making sure that there is agreement down to round-off; i.e. $[\hat{U}(t + \Delta t) - \hat{U}(t)]/\Delta t$ is compared to the spatial discretization of $L[\hat{U}(t + \Delta t)]$.

The code has been directly compared with two other time-dependent resistive MHD codes--RIPPLE IV[9] and TEMCO[10]. For a typical "tearing mode stable" equilibrium[11] excluding transport and Hall effects (but including resistivity), all three codes yield growth rates to within five percent of each other and almost identical perturbed profiles[12]. Similar agreement is obtained for the "Bessel function" equilibrium model. In a recently completed comprehensive study of resistive interchange modes in a reversed field pinch [6], results of ODRIC were compared to analytic predictions and to previously published data.

6. Helical Flux Diagnostic

In order to evaluate the effect of the perturbation, it is important to compute the magnetic field topology. Since the perturbation is of a single helicity $h = m/nk$, it is possible to define a helical flux function ψ according to

$$B_r = \frac{1}{rh} \frac{\partial \psi}{\partial \mu} \tag{26}$$

$$B_z = - \frac{1}{r} \left[\frac{\partial \psi}{\partial r} + hB_\phi \right], \tag{27}$$

where $\mu = \phi + h^{-1}z$. In the case of an axisymmetric ($m = 0$, $n \neq 0$) perturbation, h is equal to zero and $B_r = \frac{\partial \psi}{\partial z}$. The flux is computed by integrating Eq. (27) along each μ-line beginning at $r = 0$, subject to the boundary condition $\psi = 0$ at $r = 0$.

Contour plots of the helical flux are available to users on the NMFECC network. In the axisymmetric case the contours are plotted in r - z space. For $m \neq 0$ the contours are plotted in (r,μ) space using a polar representation (i.e. μ is interpreted angularly).

7. Structure of ODRIC; Subroutines and Special Features

The driver (MAIN) first calls several initializing routines:

(a) SETLAB - computes labels for output

(b) INREAD - reads in and prints out input data

(c) MESH - computes the radial mesh

(d) DRMESH - computes radial differences

(e) PERTRB - computes initial perturbation if none specified

(f) RSTART - inputs initial perturbation from disc file

(g) MULTY - scales initial perturbation

(h) ZIMAG - sets imaginary part of perturbation to zero

(i) BFM - Bessel function equilibrium

(j) PP - pitch and pressure form of equilibrium; includes tearing mode stable case

(k) INEQUL - equilibrium contained in disc file

(l) CONST - computes radius of singular surface

(m) DGNOS2 - computes maximum timesteps for explicit difference algorithm

(n) BRL - implements $r = 0$ boundary condition in difference scheme (computes top block-row of tridiagonal matrix)

(o) STCOFF - computes interior block-rows of tridiagonal matrix

(p) BCR - implements $r = r_{max}$ boundary condition in difference scheme

(q) INTPLT - plots equilibrium quantities vs radius

(r) PLOTS - plots perturbed quantities vs radius

(s) HELCAL - computes helical flux funtion; a contour plot is not supplied for non-NMFECC users

The "DO 10" loop is where the time integration is carried out; it uses the following routines:

(a) SAVOLD - stores perturbed variables at current timestep

(b) RHS - computes explicit dependencies in difference equations

(c) SETG - performs first sweep of tridiagonal solve

(d) TRSLVE - performs last sweep of tridiagonal solve

(e) SMOOTH - allows for artificial smoothing

(f) GROWTH - computes growth rates

(g) PLOTS - see above

(h) HELCAL - see above

At the conclusion of the time integration the following routines are called:

(a) GPRINT - prints growth rates vs time

(b) GPLOT - plots growth rates vs time

(c) PTERMS - plots individual physics contributions to the various equations

(d) EQNCHK - compares left and right hand sides of difference equations

(e) DLSTAR - computes $\hat{B}_r'$(real)/$\hat{B}_r$(real) and $\hat{B}_r'$(imag)/$\hat{B}_r$(imag)

(f) FIN - scales dependent variables and writes them onto disc file

There are a number of subroutines not called from MAIN, which are therefore not on the above list. They are:

(a) MLTMAT - multiples 8 by 8 matrices

(b) MLTVEC - multiplies matrix times vector

(c) ADDMAT - adds matrices

(d) SUBMAT - subtracts matrices

(e) ADDVEC - adds vectors

(f) SUBVEC - subtracts vectors

(g) INV - computes inverse of real matrix

(h) CINV - computes inverse of complex matrix

(i) FJ0 - computes $J_0(X)$

(j) FJ1 - computes $J_1(X)$

(k) ERROR - error recovery routine

(l) GRAPH - outputs y(x), y real

(m) CGRAPH - output y(x), y complex

(n) CMGROW - used to compute growth rates

(o) SAVOLD - saves dependent variables at earlier timesteps

(p) MAKEBZ - computes B_z from $\nabla \cdot \underline{B} = 0$

(q) GETR - computes mesh ratio for geometric radial mesh

(r) SPLN1D, SPL1D1, SPL1D2 - 1-D cubic splines package; (The routines SPL1D1 and SPL1D2 are modifications of those obtained from T. Jordan of Los Alamos National Laboratory [13])

(s) DU, D2U - first and second derivative functions

(t) GTERMS - computes individual physics contributions to the various equations

(u) MINV, DEC, SOL - package to invert real 8 by 8 matrix (a version of MINV in CAL exists in OMNILIB on the NMFECC system); (The routines DEC and SOL were written by A.C. Hindmarsh, et al. [14])

(v) CAXPY, CDOTC, CGECO, CGEDI, CSCAL, CSSCAL, CSWAP, SCASUM, CGEFA, ICAMAX - package to invert complex 8 by 8 matrix (these routines

are also in the SLATEC library [15])

(w) LUF - Table look-up routine

For users on the NMFECC network, output is usually in the form of plots, and the library TV80LIB is loaded. However, for those users who desire printed output, the following routines are used instead:

(a) FRAME - advances output to the next page

(b) FR80ID - creates output file

(c) TRACE - plots y vs x

(d) SETCH, KEEP80, MAPS, RCONTR, CONTUR - dummy routines

Other NMFECC features:

(a) Subroutine DROPFILE (called from MAIN) is necessary on the NMFECC system; it copies the loaded code into another file.

(b) Subroutine OPEN (called from MAIN) opens the input data file

(c) Subroutine TIMEDATE (called from INREAD) prints out the date and time.

(d) Subroutine CREATE (called from FIN) creates the output disc file.

(e) Subroutine OPEN (called from RSTART) opens the input perturbation disc file; subroutine CLOSE closes it.

(f) Subroutine OPEN (called from INEQUL) opens the equilibrium disc file.

8. PARAMETER Variables

The following variables appear in PARAMETER statements:

ICMPLX - 1 for real version, 2 for complex version

JMX - maximum number of meshwidths

IPMAX, IPMAXG - maximum number of time levels at which growth rates are saved for (prints, plots), respectively

JMX1 - maximum number of meshpoints

JJJ, III - limits on r and μ meshes for helical flux diagnostic

NCONTF - number of contours in helical flux diagnostic

9. COMMON Variables

Block MATS

A, B, C, D, E, G - correspond to variables in Eqs. (23-25);

A and B need not be stored at each meshpoint simultaneously

U - $\hat{U} = (\hat{B}_r, \hat{B}_\phi, \hat{T}_e, \hat{V}_r, \hat{V}_\phi, \hat{V}_z, \hat{\rho}, \hat{T}_i)$ (Eq. (21))

R - radial mesh (r)

BTO, BTOP - $B_{\phi o}$, $B_{\phi o}'$ (Eq. (18))

BZO, BZOP, BZO2P - B_{zo}, B_{zo}', B_{zo}'' (Eq. (18))

TO, TOP, TOPP - T_o, T_o', T_o''

RHOO, RHOOP - ρ_o, ρ_o' (Eq. (17))

(NOTE: ρ_o' must be zero; certain areas of code assume uniform equilibrium density)

ETAO, ETAOP - η_o, η_o' (NOTE: $\eta = \eta_o$) (Eq. (17))

L - identically equal to 1

F - $\underline{k} \cdot \underline{B}_o$, where $\underline{k}$ is the wave number vector

BCE - $(B_j - C_j E_{j-1})^{-1}$

P, PG - growth rate prints/plots; the first subscript is the time level; the second subscript is the variable number; the first and second entries of the third subscript are the real and imaginary parts as computed using the Buneman routine, and the third entry is the real growth rate computed using successive quotients; the fourth subscript allows two entries corresponding to the real and imaginary variable parts for a complex case.

SDEV - standard deviation of growth rate divided by mean

PRTMES, PTIMEG - array of time values for growth rate prints/plots

NAVE - number of meshpoints at which growth rate is computed successfully

OMEGA - artificial smoothing coefficient (multiplies undivided second difference)

BZ, BZP - $\hat{B}_z$, $\hat{B}_z'$ (Eq. (21))

EQN - forces perturbed variables to maintain a constant profile while growing at a specified rate (PGRO)

DR - $\frac{1}{2}(r_{j+1}-r_{j-1})$ if $0 < j < J$

r_1-r_0 if $j=0$

r_J-r_{J-1} if $j=J$

DRP - $r_{j+1} - r_j$

DRM - $r_j - r_{j-1}$

MULT - multiplies input perturbation

IMUS - switches for viscosity

U0 - values of perturbed variables at previous times

Block CONS

M - m (Eq. (21))

GAMMA - γ (Eqs. (2-3))

RW - wall radius

JMAX - J

C0 - determines magnitude of initial perturbation

MODEL - equilibrium model

C1 - equilibrium magnitude with Bessel function model

KSTARE - $\kappa_{e\perp}$ (Eq. (9))

KPARE - $\kappa_{e\parallel} - \kappa_{e\perp}$ (Eq. (9))

KSTARI - $\kappa_{i\perp}$ (Eq. (10))

KPARI - $\kappa_{i\parallel} - \kappa_{i\perp}$ (Eq. (10))

ALPHA - $\left[(m/r_s)^2 + (nk)^2\right]^{1/2}$, where r_s is the singular surface radius

KZ - nk (Eq. (21))

BETA - 1 (not used)

DT - timestep (Δt)

ITIME - cycle number

IPRINT, IPLOTG - frequency of adding entries to growth rate print/plot tables

IPLOT - frequency of plotting dependent variables

IHPLOT - frequency of plotting helical flux contours

NU - multiplies argument in Bessel function equilibrium

ITMAX - number of timesteps

S - $S = \tau_R/\tau_A$ ($\eta = S^{-1}$; see Eq. (7))

PLTIME, PRTIME, PLTMEG, SMTIME - used in conjunction with IPLOT, IPRINT, IPLOTG, ISMOTH

IP, IPG - number of time entries in growth rate prints/plots

TOIN - T_o at wall (Eq. (17))

RHOOIN - ρ_o (Eq. (17))

INFILE, OTFILE - input/output perturbation files

ISMOTH - frequency of applying artificial smoothing

ID - run identification

ICOPT - chooses initial velocity perturbation for M = 1

MSHTPE - geometric or uniform mesh

J1, J2, R1, R2 - used to define geometric radial mesh

AA, BB, CC - used to define Suydam parameter for tearing mode stable equilibrium

JRR - mesh index of singular radius

GHALL - determines fraction of equilibrium pressure in electrons

WCTAI - Hall parameter ν (Eq. (8))

MUSTAR - viscosity coefficients (Eq. (11))

ETA0IN - η_o (Eq. (7))

RR - singular radius

PGRO, GROFAC - used in conjunction with EQN = "OFF"

NU0 - $\nu/4$ (Eq. (8))

TIME - t

IHALLT - decides if Hall terms are to be included in temperature equation

ZERMAG - used to set imaginary part of initial perturbation to zero

ECHECK, EPSMIN - used to perform consistency check on equations

GROUT - switch for outputting growth rate diagnostics

TERMS - switch for plotting individual physics terms

FOHMH - switch for inclusion of Ohmic heating

NUEI - ν_{ei} (Eqs. (2-3))

DSKOLD - describes format of input disc file

YE, YI - $\kappa_{e\wedge}$, $\kappa_{i\wedge}$ (Eqs. (9-10))

GAMMAT - γ_T (Eq. (14))

Block KKPLOT

IUN - logical unit for output (100 for plots on NMFECC network)

ISKIP - frequency of meshpoints included in output

Block CHRS

LABP, LAB - output labels

Block MSPL

SPL1, SPL2 - arrays containing information fed into 1-D spline package

W, T1, T2, T3 - extra storage for 1-D spline package

RSPL - radial nodes for disc input to be splined

Block CTERMS

BRVXB through TICOLL - individual contribution to the equations for $\hat{U}$

ULHS, URHS, UDIF - left side ($\Delta\hat{U}/\Delta t$), right side (physics), and relative difference of equations for $\hat{U}$

TLHS, TRHS, TDIF - temporary values of ULHS, URHS, UDIF

AMX, BMX - suprema of UDIF over mesh

10. Input

Input data is read in subroutine READIN. All of of the input variables are in COMMON and have been described in the previous section; a more comprehensive description appears in the comments at the beginning of the code. The equilibrium may be input in binary form in subroutine INEQUL. The initial perturbation may be read in binary form in subroutine RSTART.

11. Output

The final perturbed state is normalized and output into a binary disc file. The remaining output appears in print format, although users of the NMFECC network may instead use TV80LIB and obtain plotted output. At the beginning of the run, the date and time (NMFECC only) are printed out, followed by the values of all of the input variables, followed by the nondimensional wave number $\alpha = \left[(m/r_s)^2 + (nk)^2\right]$, the singular radius r_s, and the maximum diffusive and advective timesteps (TRMIN and TAMIN) for an explicit difference scheme. Various equilibrium variables ($B_{\phi o}$, $B'_{\phi o}$, B_{zo}, B'_{zo}, T_o, T'_o, ρ_o, ρ'_o, η_o, η'_o, $F = \underline{k} \cdot \underline{B}_o$) come next, followed by the perturbed variables; in each case

the radially dependent variables are printed along with the corresponding radius; the integer appearing to the left of the radius is generally the mesh index plus one. At later times printed information includes the perturbed variables in the above format along with growth rate information at various times. The latter is printed twice (in differing formats; in the second instance the data would normally be in plot mode on the NMFECC network.) In the first instance the real and complex growth rates of each variable averaged over radius are printed out, along with the relative standard deviation and the number of meshpoints which admit a valid growth rate diagnostic. The quantity "P" stands for the real growth rate, and "GAMMA + i OMEGA" is the complex growth rate. In the second instance only the growth rates themselves are printed out. Both real and complex growth rate diagnostics are included, since the complex growth rate calculation breaks down in the real limit. Following the growth rate information are the quantities $\hat{B}_r'(\text{real})/\hat{B}_r(\text{real})$ and $\hat{B}_r'(\text{imag})/\hat{B}_r(\text{imag})$ vs r. In the case of the geometric mesh option these quantities are printed out both over the whole radial domain and in the neighborhood of the singular surface. The final diagnostic is the CPU time (NMFECC only).

12. Test run 1 (real calculation)

Test run 1 uses an equilibrium of Robinson [11] which is stable to both Suydam and tearing modes[16,17]; the aim is to find the fastest growing resistive interchange mode[17]. The equilibrium is described in terms of the Suydam parameter C_1 (which must be less than 0.125 for stability) and the pitch function $\mu = rB_z/B_\phi$, as follows:

$$\mu = 2(1 - .125r^2 - .0025r^4) \tag{28}$$

$$C_1 = \frac{- p'}{2rB_z^2}\left(\frac{\mu}{\mu'}\right)^2 = 0.1, \tag{29}$$

where $p = \rho T$ and $\rho = 1$. The poloidal and toroidal mode numbers m and nk are equal to 1 and -1.8, respectively. The initial conditions consist of parabolic $\hat{V}_r$ and $\hat{V}_\phi$ profiles. A complete list of input parameters is given in Table 2.

The printed output shows, in addition to information at $t = 0$, the perturbed variables and growth rate evolution at the end of the run. The (real) growth rate converges to a value of approximately 1.649×10^{-3}. A plot of $\hat{V}_r$ vs r appears in Fig. 1.

13. Test run 2 (complex calculation)

This case is similar to test run 1, except that the Hall, thermal force, transverse thermal conductivity and FLR terms, which allow the real and imaginary parts of perturbations to couple, are nonzero. In addition, the parallel viscosity is 100 times greater and the number of timesteps is 300 as compared to 500.

The resulting (complex) growth rate is equal to $2.01 \times 10^{-2} + 1.18 \times 10^{-2}$ i. Plots of $\hat{V}_r$ vs r appear in Figs. 2 and 3.

References

[1] R.C. Grimm, J.M. Greene and J.L. Johnson, Methods in Computational Physics 16 (Academic Press, New York, 1976), 253.

[2] R. Gruber, et al., Comput. Phys. Commun. 21, 323 (1981).

[3] J.A. Dibiase and J. Killeen, J. Comput. Phys. 24, 158 (1977).

[4] A.I. Shestakov, J. Killeen and D.D. Schnack, J. Comput. Phys. 46, 69 (1982).

[5] S.I. Braginskii, in Reviews of Plasma Physics 1 (Consultants Bureau, New York, 1965), 205.

[6] A.A. Mirin, N.J. O'Neill, J. Killeen, R.J. Bonugli and M.J. Ellis, "Linear Studies of Resistive Interchange Modes in a Cylindrical Reversed Field Pinch," to appear in Phys. Fluids.

[7] R.D. Richtmyer and K.W. Morton, Difference Methods for Initial Value Problems (Wiley, New York, 1967), 198.

[8] O. Buneman, J. Comput. Phys. 29, 295 (1978).

[9] J. Killeen, in Physics of Hot Plasmas (Plenum, New York, 1970), 202.

[10] A.A. Mirin, N.J. O'Neill and A.G. Sgro, Proc. U.S.-Japan Theory Workshop on 3-D MHD Studies, (Oak Ridge, 1984), 88.

[11] D.C. Robinson, Nucl. Fusion 18, 939 (1978).

[12] J. Killeen and A.A. Mirin, Bull. Amer. Phys. Soc. 27, 941 (1982).

[13] T. Jordan, private communication.

[14] A.C. Hindmarsh, L.J. Sloan and P.F. Dubois, "DEC/SOL: Solution of Dense Systems of Linear Algebraic Equations," Lawrence Livermore National Laboratory Report UCID-30137, Rev. 1 (1978).

[15] Sandia, Los Alamos, Air Force Weapons Laboratory Technical Exchange Committee (SLATEC) Library, distributed through National Energy Software

Center, Argonne National Laboratory, Argonne, Illinois.

[16] B.R. Suydam, in Peaceful Uses of Atomic Energy 31 (Proc. Int. Conf., United Nations, Geneva, 1958), 157.

[17] H.P. Furth, J. Killeen and M.N. Rosenbluth, Phys. Fluids 6, 459 (1963).

TABLE 1. NORMALIZATION OF VARIABLES

Variable	Normalization Factor
r	a
ρ	$\bar{\rho}$
B	$\bar{B}$
v	$v_A = \bar{B}\ (4\pi\bar{\rho})^{-1/2}$
t	$\tau_A = a/v_A$
$p = \rho T$	$\bar{p} = \bar{B}^2/8\pi$
T, T_e, T_i	$\bar{T} = \bar{p}/\bar{\rho}$
η	$\bar{\eta} = 4\pi a^2/c^2\tau_A$
$\underline{\kappa}_e$, $\underline{\kappa}_i$	$\bar{\kappa} = \bar{\rho}a^2/\tau_A$
μ_j	$\bar{\mu} = \bar{\rho}a^2/2\tau_A$
ν_{ei}	$\bar{\nu} = 1/\tau_A$

The quantities a, $\bar{\rho}$ and $\bar{B}$ may be independently specified. The resistive diffusion time τ_R is equal to $4\pi a^2/\eta_\perp c^2$, where $\eta_\perp$ is the perpendicular resistivity. The perturbed variables ρ_1, T_{e1}, T_{i1}, V_{r1}, $B_{\phi 1}$, B_{z1} are further normalized by a factor of i.

TABLE 2. INPUT FOR TEST RUNS 1 AND 2

The values for test run 2 (when different) appear in parentheses.

ID =	#1 (#2)
INFILE =	NONE
OTFILE =	ODOUTTP1 (ODOUTTP2)
IUN =	3
ISKIP=	15
M =	1
KZ =	-1.8
GAMMA =	1.6667
S =	1000.
KSTARE=	1.45×10^{-7}
KPARE =	0.449
KSTARI =	2.67×10^{-6}
KPARI =	0.0129
FOHMH =	1.
WCTAI =	0.
GHALL =	0
IHALLT =	0
MUSTAR =	0.951 (95.1), 1.2×10^{-6}, 4.8×10^{-6}, 0.(0.1) 0.(0.2)
NUEI =	1.08×10^{-5}
YE =	0 (.0256)
YI =	0 (.0249)
GAMMAT =	0 (.31)
IMUS =	1,1,1
RW =	3
DT =	10

ZERMAG =	NO	
DSKOLD =	YES	
OMEGA =	0.	
EQN =	ON	
MULT =	1.	
PGRO =	0.	
C0 =	1.	
ITMAX =	500	(300)
IPRINT =	20	
IPLOT =	500	(300)
IHPLOT =	0	
IPLOTG =	20	
ISMOTH =	1	
ICOPT =	A	
MSHTPE =	GEO	
J1 =	300	
J2 =	1200	
R1 =	2.1	
R2 =	2.5	
JMAX =	1500	
MODEL =	TMS	
AA =	0.1	
BB =	0.	
CC =	0.	
TOIN =	0.1	
RHOOIN =	1.	
ETAOIN =	1.	

GROUT -	YES
TERMS -	NO
ECHECK -	NO
EPSMIN -	0.01

Figure Captions

Figure 1. Plot of $\hat{V}_r$ vs r for test run 1.

Figure 2. Plot of the real part of $\hat{V}_r$ vs r for test run 2.

Figure 3. Plot of the imaginary part of $\hat{V}_r$ vs r for test run 2.

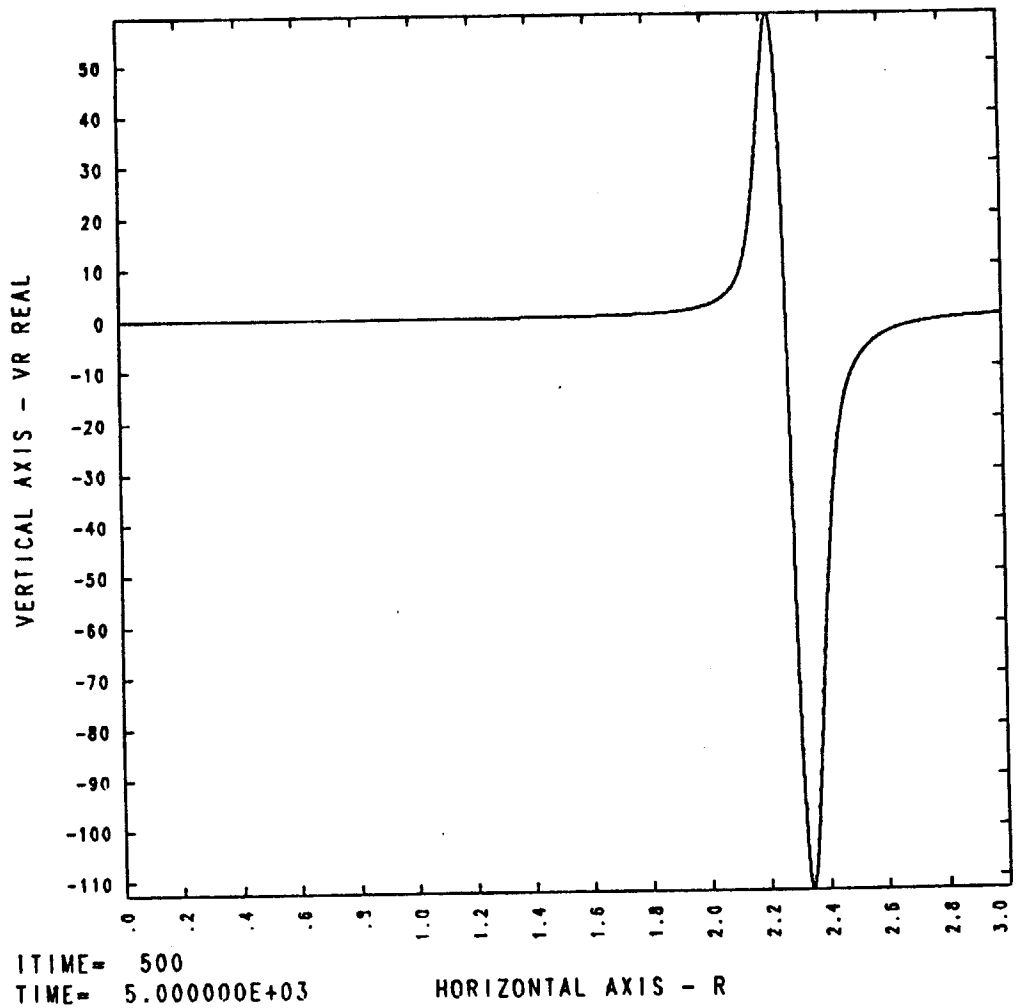
VERTICAL AXIS - VR REAL
50
40
30
20
10
0
-10
-20
-30
-40
-50
-60
-70
-80
-90
-100
-110
.0
.2
.4
.6
.8
1.0
1.2
1.4
1.6
1.8
2.0
2.2
2.4
2.6
2.8
3.0
ITIME= 500
TIME= 5.000000E+03
HORIZONTAL AXIS - R

Figure 1

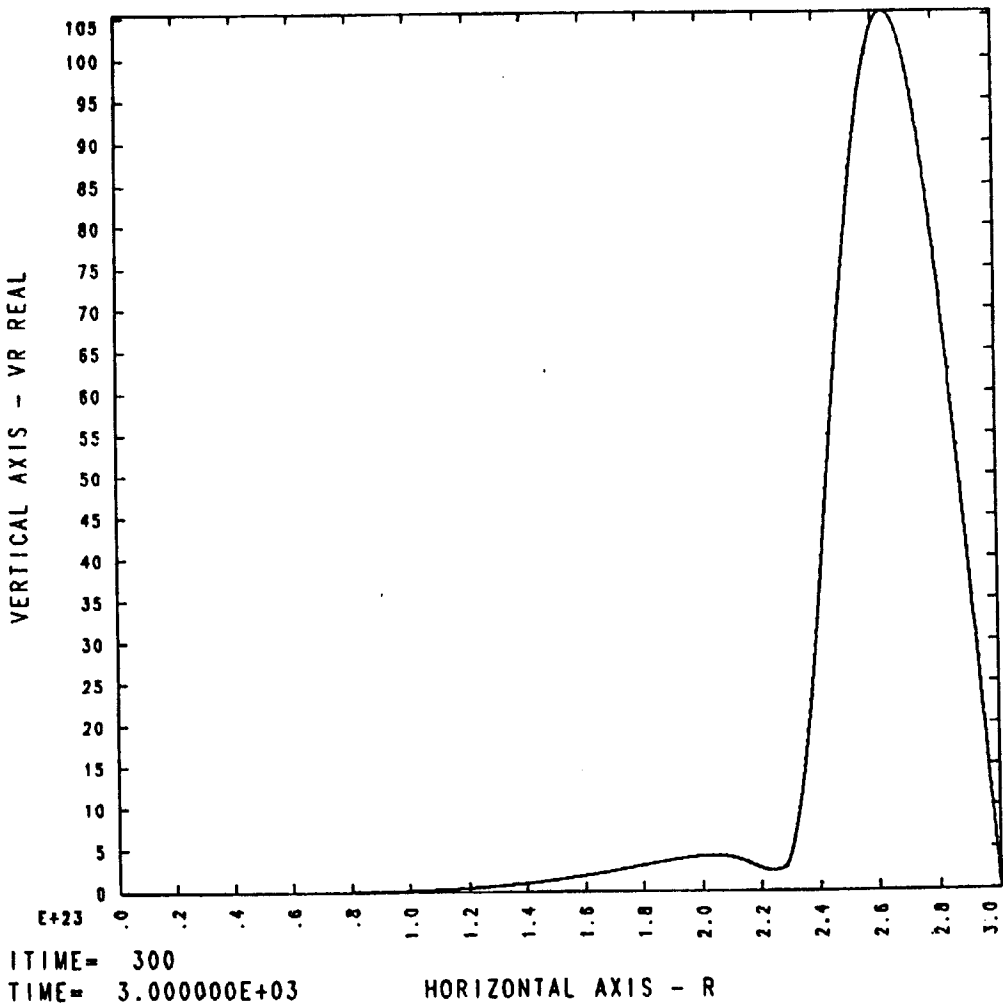
VERTICAL AXIS - VR REAL
105
100
95
90
85
80
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
E+23
.0
.2
.4
.6
.8
1.0
1.2
1.4
1.6
1.8
2.0
2.2
2.4
2.6
2.8
3.0
ITIME= 300
TIME= 3.000000E+03
HORIZONTAL AXIS - R

Figure 2

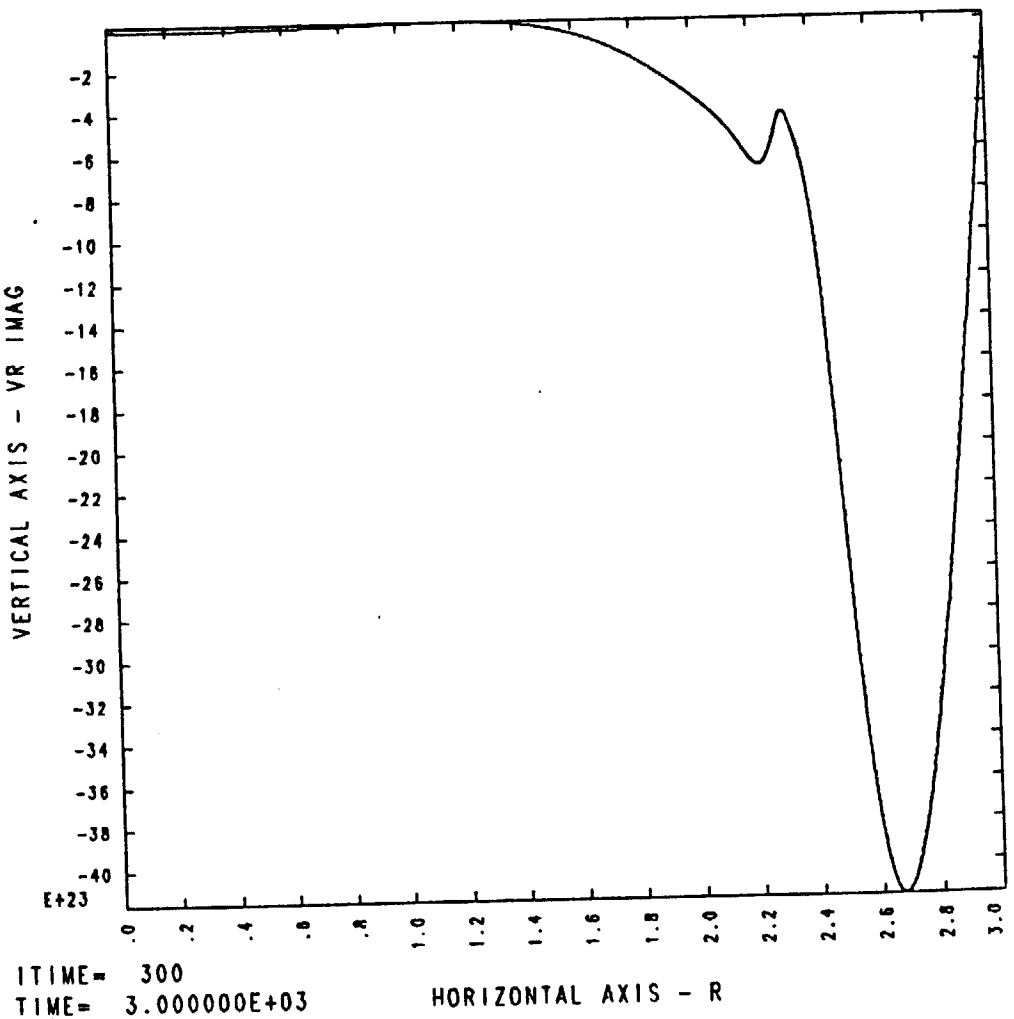
VERTICAL AXIS - VR IMAG
-2
-4
-6
-8
-10
-12
-14
-16
-18
-20
-22
-24
-26
-28
-30
-32
-34
-36
-38
-40
E+23
.0
.2
.4
.6
.8
1.0
1.2
1.4
1.6
1.8
2.0
2.2
2.4
2.6
2.8
3.0
ITIME= 300
TIME= 3.000000E+03
HORIZONTAL AXIS - R

Figure 3